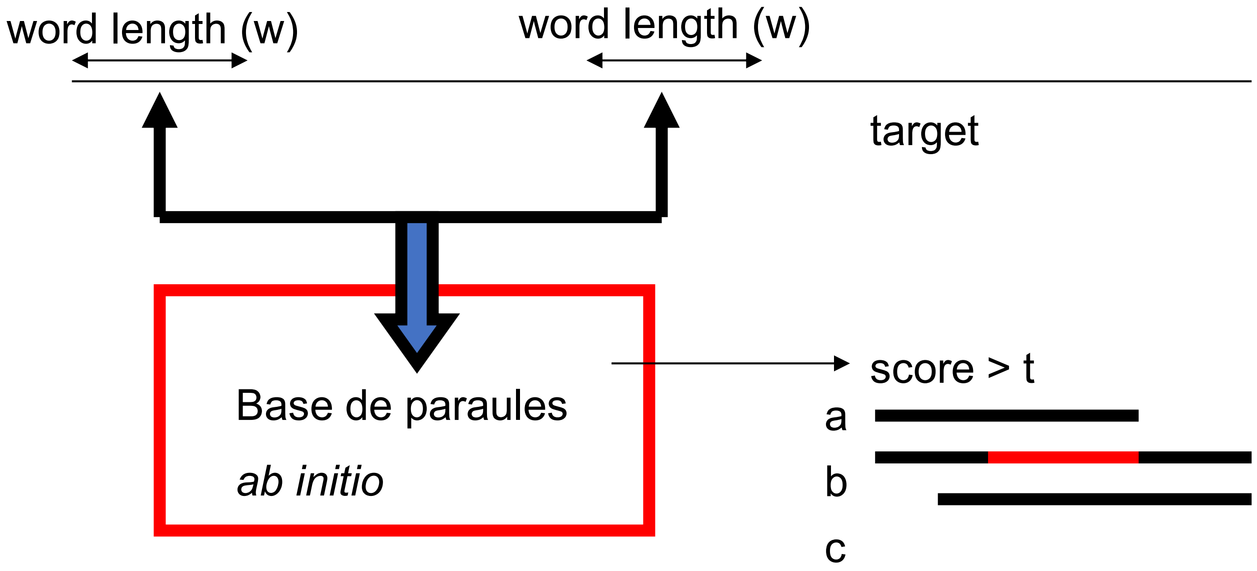
Exercise 2

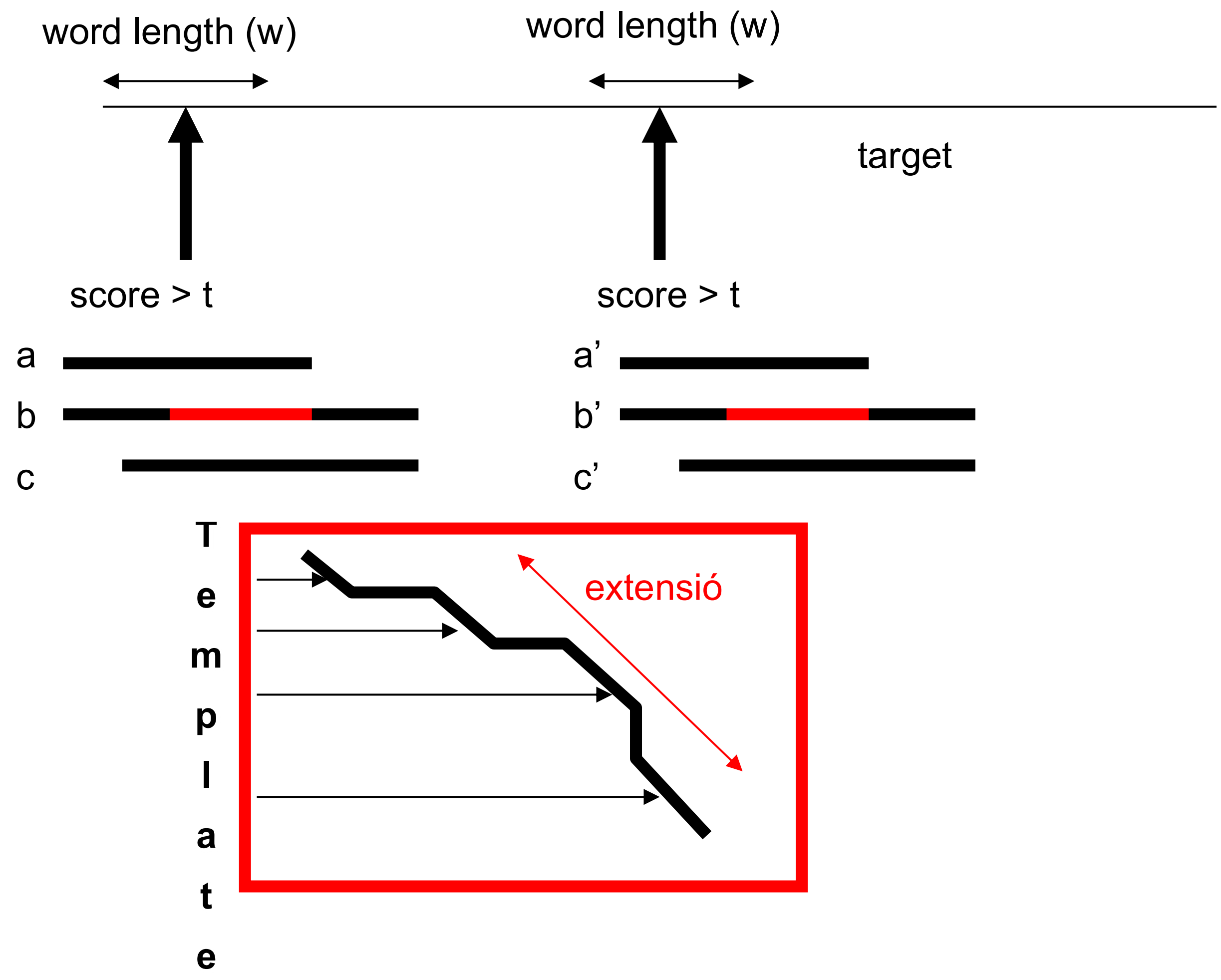
Search for similar sequences (PSI-BLAST and HMM.)

We propose the following problem: We have the sequence of a protein of which we know nothing, neither its function nor structure. This is named protein-problem or target. We want to know if there is a protein with known structure and/or function that shares a common ancestor with the protein problem (definition of homology). Our hypothesis is that if we find a sequence that we can align with the sequence of the problem, and the total number of aligned residues is unlikely to occur by chance (p-value), then the protein problem shares a common ancestor with the protein found (i.e. both are homologs). Consequently, the function and structure of the protein problem is the same as its homolog.

Solutions with BLAST (exercice 2.1)

First, we create a database of words with the sequence of the target. Then, we search identical words in a database of words. This database is made with all the sequences of known proteins. Finally join all words (with gaps, if necessary) to reconstruct the maximum length of the sequence of the target aligned with one of the sequences of the database.





The expected value of a particular score, S, is:

$$\boxed{E(S) \approx \exp\left(-NmnKe^{-(S-\mu)}\right)}$$

And the probability of obtaining a value greater than S:

$$P(x > S) \approx 1 - e^{-E(S)}$$

$$S > T + \frac{\log(m \bullet n)}{\lambda}$$

Where n and m are the number of residues of the sequences being compared, N is the size of the database, K and λ are parameters and μ is the average of scores.

The scores, S, are the sum of the scores of residues aligned. This scores are taken from a substitution matrix. If you glutamic (E) is aligned with aspartic (D) the score of the aligned pair is M (E, D), where M comes from the matrix of substitutions in 20x20.
One of the most widely used matrices is Blosum62. This has been obtained by algned blocks of similar sequences, presumably indicating the evolutionary changes of the residues of a protein.

Examples:

```
➢ blastp -query [target_fasta_format]-db [database]-out [output]
```

TUTORIAL (step1):

Download "exercise_2.tar" from aula global and unpack

Within "exercise_2" you will find the subdirectory BLAST. Within this there is the sequence problem named "target.fa" .

To look for proteins of known structure similar to the target protein, try

```
> blastp –query target.fa -out target_pdb.out -db /mnt/NFS_UPF/soft/databases/blastdat/pdb_seq
```

You can see the result of the search in the output file target_pdb.out.

<u>Solutions with PSI-BLAST</u>

The problem of using Blosum62 is that the substitution matrix is generic, therefore we always use the same matrix for all our searches. If we were able to use a matrix with specific substitutions for a given family of proteins, this would be more specific for each position along the sequence and it would improve the result of the search. This matrix is called PSSM (Position Specific Substitution Matrix). But, how can we obtain this matrix?

PSI-BLAST is an iterative method, it looks for sequences similar to the target protein and with the alignments obtained it recalculates a new substitution matrix. The matrix is stored and used to add more similar sequences.
Iteration 0> Target + Blosum62 -> [similar sequences] $_0$ -> matrix (PSSM) + Target
Iteration 1> Target + (PSSM) $_0$ -> [similar sequences] $_1$
Iteration 2> Target + (PSSM) $_1$ -> [sequences similar] $_2$
......

Example:

```
➢ psiblast -query [target_fasta_format] -db [database] –num_iterations
[number of iterations] -out [output]
```

TUTORIAL (step 2)

Using the previous example:

```
> psiblast -query target.fa –num_iterations 5 –out target_pdb_5.out -db
/mnt/NFS_UPF/soft/databases/blastdat/pdb_seq
```

Results per iteration are indicated by the word “Round”. You can search this word in the output file to compare the results at different iteration rounds.

But this raises a problem: the database "pdb_seq" contains only sequences with known structure. Then, the result is biased towards those families with higher number of known structures (i.e. that have been crystallized or studied by NMR) instead of incorporating the evolutionary information about substitutions, deletions and/or insertions.

To solve this problem we need a database with all protein sequences. This database is NR. We may also use those sequences for which we know its function (SwissProt), since this is also a database very big with very small bias. However, we need to store the PSSM matrix and use it in the next search. The option to store the PSSM is "-C", and it works as follows:

```
psiblast -query target.fa -num_iterations 5   -out_pssm target_sprot5.pssm -out
target_sprot_5.out -db
/mnt/NFS_UPF/soft/databases/blastdat/uniprot_sprot.fasta
```

Next, we apply the new PSSM to search among sequences with known structure (in pdb_seq):

```
psiblast -db /mnt/NFS_UPF/soft/databases/blastdat/pdb_seq -in_pssm
target_sprot5.pssm -out target_pdb_sprot5.out
```

Note that we no longer iterate in pdb_seq. This is because if we iterate a new PSSM matrix the evolutionary significance would be biased again.

Other options of "psiblast" are:

-num_descriptions/max_target_seqs:
Maximum number of aligned sequences in output

-evalue:
E-value threshold to see output sequences (default 10)
-inclusion_ethresh:
E-value threshold to include sequences in the PSSM (default 0.002)

To see all options by doing "psiblast -help"

Finally, to use a specific alignment of sequences we need a special format. This is possible with the ClustalW program.

Example:

```
clustalw seq_set.fa
```

TUTORIAL (step 3)

We extract the sequences with FetchFasta.pl using a list, with a format taken from the blast output. The list is a file that YOU HAVE TO GENERATE, for example using “gedit” or “gvim”, and it may be saved with a name such as file.list. For example, the format of the file “file.list” with sequences from the database “sprot.fas” looks like this:

```
ORC1_DROME (O16810) ORIGIN RECOGNITION COMPLEX SUBUNIT 1 (DMORC1).    380   e-105
ORC1_SCHPO (P54789) ORIGIN RECOGNITION COMPLEX SUBUNIT 1.             295   2e-79
ORC1_CANAL (O74270) ORIGIN RECOGNITION COMPLEX SUBUNIT 1.             223   1e-57
ORC1_YEAST (P54784) ORIGIN RECOGNITION COMPLEX SUBUNIT 1 (ORIGIN...   189   1e-47
ORC1_KLULA (P54788) ORIGIN RECOGNITION COMPLEX SUBUNIT 1.             179   1e-44
CC18_SCHPO (P41411) CELL DIVISION CONTROL PROTEIN 18.                 115   2e-25
CC6_YEAST (P09119) CELL DIVISION CONTROL PROTEIN 6.                    87   8e-17
YPZ1_METTF (P29570) HYPOTHETICAL 40.6 KDA PROTEIN (ORF1').             60   1e-08
YPV1_METTF (P29569) HYPOTHETICAL 40.7 KDA PROTEIN (ORF1).              60   1e-08
SIR3_YEAST (P06701) REGULATORY PROTEIN SIR3 (SILENT INFORMATION ...    47   1e-04
G6PI_OENME (P54243) GLUCOSE-6-PHOSPHATE ISOMERASE, CYTOSOLIC (GP...    31   6.6
```

If these sequences are from swissprot, then use the following command:

```
>perl /mnt/NFS_UPF/soft/perl-lib/FetchFasta.pl –i file.list   –o
file.fasta –d /mnt/NFS_UPF/soft/databases/blastdat/uniprot_sprot.fasta
```

We need to obtain the sequence alignment of these sequences plus the target sequence. DO NOT FORGET TO INCLUDE THE TARGET SEQUENCE. The target sequence has to be included as the first sequences in the file, because is required to indicate the position of the residues in the PSSM.

This can be done by running:

```
>cat target.fa >pssm.fasta
```

```
>cat file.fasta>>pssm.fasta
```

We'll run clustalw to get an alignment, but we need to get the results in FASTA format so we use the command:

```
>clustalw -INFILE=pssm.fasta -OUTFILE=clustalw.pssm -OUTPUT=FASTA
```

Then, we use the option **–in_msa** with the output file

```
>psiblast –in_msa clustalw.pssm –out target_pdb_specific.out -db
/mnt/NFS_UPF/soft/databases/blastdat/pdb_seq
```

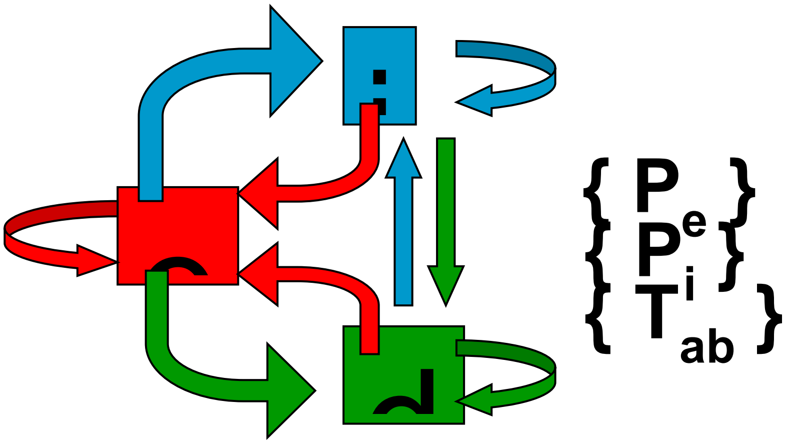
QUESTIONS FROM THE TUTORIAL

Now we can compare all the results and answer the following questions:

1) Why are the e-values different in *target_pdb.out* than in the fifth iteration in *target_pdb_5.out*?
2) Why do we need to run blastpgp with uniprot_sprot.fasta before searching in pdb_seq?
3) When obtaining the file *target_pdb_sprot5.out* why we didn't run 5 iterations as before?
4) Search in the SCOP database with the PDB code of the best match of the target sequence. Do all the files *target_pdb_specific.out, target_pdb_sprot5.out, target_pdb_5.out* and *target_pdb.out* produce the same result?
5) Can you use the file target_sprot5.out to obtain the name of the fold in SCOP? Why?
6) What are the folds of the following sequences?
   a. *problem1/serc_myctu.fa*
   b. *problem2/p72_mycmy.fa*
   c. *problem3/lip_staau.fa*
   d. *problem4/orc1_human.fa*

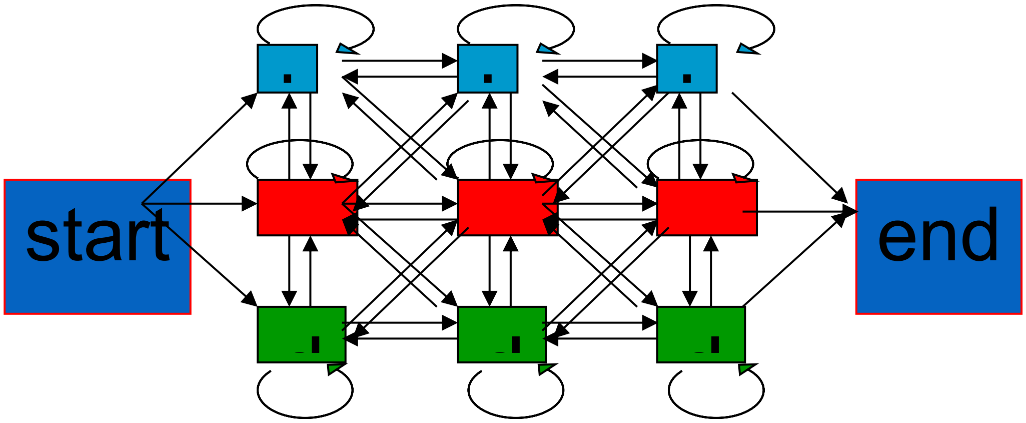
Solution by HMM (exercice 2.2)

Hidden Markov Models (HMM) are based on the probability of a having a sequence. A sequence is generated with a hidden model of sequence emission. The model includes the generation of gaps and the emission under two different sates: insertion state and main state



Besides the probability to emit one residue in the states of insertion (Pi) or main (Pe), we also use the probability to change from one to other state. A deletion state cannot emit residues (T).

A sequence can be built using several states of emission, insertion and deletion. In addition, some residues appear several times, n, on the same state. Let be a selection of states with a particular order, π, and in each state the emission of some residues, A, such that at the end the final sequence is $\Theta=A_1A_2A_3...A_L$.



Then the probability of finding the sequence with the path π, is $P(\Theta, \pi |M)$, being M the set of parameters (Pi, Pe and T), and the probability of having sequence Θ is $P(\Theta | M)$:

$$P(\Theta,\pi \mid M) = \prod t_{ji} \prod e_{iX}^{n(i,X,\pi,\Theta)}$$

$$P(\Theta \mid M) = \sum_{\pi} P(\Theta,\pi \mid M)$$

To generate the parameters M of a particular family of sequences we need a previous alignment, named seed and we use the program hmmbuild (NOTE: remember to activate the aliases, as in the exercise with BLAST, otherwise the path to the program hmmbuild won't be known):

- **hmmbuild [model_HMM] [alignment]**

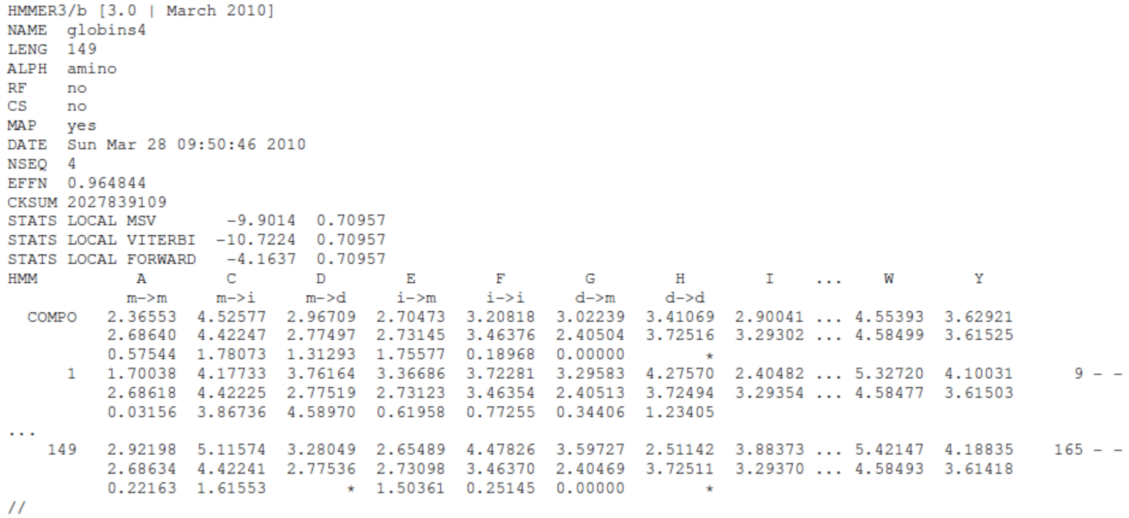
The alignment has to be in STOCKHOLM format, like **globins4.sto**. You will find the required files in the folder HMMER within the directory of exercise_2. We run this as an example:

>hmmbuild globins4.hmm globins4.sto

The file globins4.sto contains an alignment with the STOCKHOLM format, classical of PFAM database. We need a script to transform this format into clustalw format and back, this will allow us to most changes of format. The script "aconvertMod2.pl" allows us to change several alignment formats

Now you can open the file globins.hmm to check each column ad row. Each position ha the logarithm of the probability of emission of a residue. The Aa order is defined in two specific rows of the header (HMM). For each position we have probabilities on two different states (insertion and main), then we have a third row per position with the probabilities of transitions

MODEL PARAMETERS



```
HMMER3/b [3.0 | March 2010]
NAME  globins4
LENG  149
ALPH  amino
RF    no
CS    no
MAP   yes
DATE  Sun Mar 28 09:50:46 2010
NSEQ  4
EFFN  0.964844
CKSUM 2027839109
STATS LOCAL MSV       -9.9014  0.70957
STATS LOCAL VITERBI  -10.7224  0.70957
STATS LOCAL FORWARD   -4.1637  0.70957
HMM          A        C        D        E        F        G        H        I    ...    W        Y
            m->m     m->i     m->d     i->m     i->i     d->m     d->d
  COMPO   2.36553  4.52577  2.96709  2.70473  3.20818  3.02239  3.41069  2.90041 ... 4.55393  3.62921
          2.68640  4.42247  2.77497  2.73145  3.46376  2.40504  3.72516  3.29302 ... 4.58499  3.61525
          0.57544  1.78073  1.31293  1.75577  0.18968  0.00000        *
      1   1.70038  4.17733  3.76164  3.36686  3.72281  3.29583  4.27570  2.40482 ... 5.32720  4.10031      9 - -
          2.68618  4.42225  2.77519  2.73123  3.46354  2.40513  3.72494  3.29354 ... 4.58477  3.61503
          0.03156  3.86736  4.58970  0.61958  0.77255  0.34406  1.23405
...
    149   2.92198  5.11574  3.28049  2.65489  4.47826  3.59727  2.51142  3.88373 ... 5.42147  4.18835    165 - -
          2.68634  4.42241  2.77536  2.73098  3.46370  2.40469  3.72511  3.29370 ... 4.58493  3.61418
          0.22163  1.61553        *  1.50361  0.25145  0.00000        *
//
```

The columns for A C D etc. are the values to score the probabilities of the residues Ala, Cys, Asp etc. in the main state (first row) or insertion state (second row). The third row is for transition scores (see section 5 of the UserGuide manual HMMER3.0). The COMPO row refers to average scores for all residues. Next two rows refer to the BEGIN state (emissions and transitions, respectively). The END state is defined by LENG, that indicates the last state of the model.

Now we can use this model to search for other sequences that would fit with the model, obtaining good p-values:

- **hmmsearch (options) [model_HMM] [database] > [output]**

In our example, we can search for sequences with known structure fitting with the family of globins:

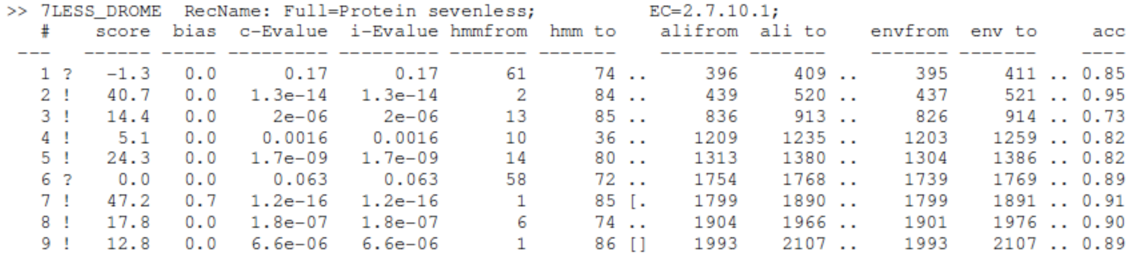
>hmmsearch globins4.hmm /mnt/NFS_UPF/soft/databases/blastdat/pdb_seq > globins_pdb.out

The results are sequences sorted by E-value, as in BLAST. But, we can also search a domain within a single sequence. Run the following commands:

>hmmbuild fn3.hmm fn3.sto

>hmmsearch fn3.hmm 7LES_DROME>fn3.out

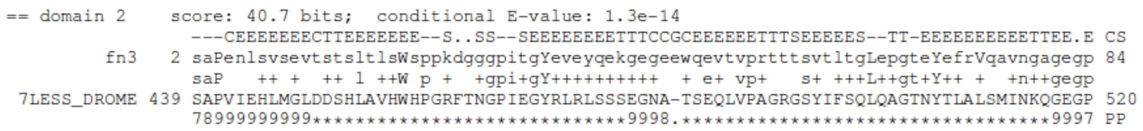
The file 7LES_DROME only contains one sequence, therefore only one E-value is retrieved. Nevertheless, the HMM profile finds several domains within this sequence. The output looks like this:



```
>> 7LESS_DROME  RecName: Full=Protein sevenless;          EC=2.7.10.1;
   #    score  bias  c-Evalue  i-Evalue hmmfrom  hmm to    alifrom  ali to    envfrom  env to     acc
 ---   ------ ----- --------- --------- ------- -------    ------- -------    ------- -------    ----
   1 ?   -1.3   0.0      0.17      0.17      61      74 ..     396     409 ..     395     411 .. 0.85
   2 !   40.7   0.0   1.3e-14   1.3e-14       2      84 ..     439     520 ..     437     521 .. 0.95
   3 !   14.4   0.0     2e-06     2e-06      13      85 ..     836     913 ..     826     914 .. 0.73
   4 !    5.1   0.0    0.0016    0.0016      10      36 ..    1209    1235 ..    1203    1259 .. 0.82
   5 !   24.3   0.0   1.7e-09   1.7e-09      14      80 ..    1313    1380 ..    1304    1386 .. 0.82
   6 ?    0.0   0.0     0.063     0.063      58      72 ..    1754    1768 ..    1739    1769 .. 0.89
   7 !   47.2   0.7   1.2e-16   1.2e-16       1      85 [.    1799    1890 ..    1799    1891 .. 0.91
   8 !   17.8   0.0   1.8e-07   1.8e-07       6      74 ..    1904    1966 ..    1901    1976 .. 0.90
   9 !   12.8   0.0   6.6e-06   6.6e-06       1      86 []    1993    2107 ..    1993    2107 .. 0.89
```

This shows the E-values of the domains: C-Evalue is the significance of matching the domains under the condition that the sequence is a member of the family defined by the HMM profile, I-Evalue is the independent E-value showing he significance of the domain within the whole set of sequences of the searched database (in this case there is only one sequence, so C-Evalue and I-Evalue coincide). The rest of columns indicate the residues for starting and ending the alignment of the profile and the sequence.

Finally, the alignment between sequence and profile is reported:



```
== domain 2    score: 40.7 bits;  conditional E-value: 1.3e-14
                 ---CEEEEEEECTTEEEEEEE--S..SS--SEEEEEEEETTTCCGCEEEEEETTTSEEEEES--TT-EEEEEEEEEETTEE.E CS
         fn3   2 saPenlsvsevtstsltlsWsppkdgggpitgYeveyqekgegeewqevtvprtttsvtltgLepgteYefrVqavngagegp 84
                 saP   ++ +  ++ l ++W p +  +gpi+gY++++++++++  + e+ vp+   s+ +++L++gt+Y++ +  +n++gegp
 7LESS_DROME 439 SAPVIEHLMGLDDSHLAVHWHPGRFTNGPIEGYRLRLSSSEGNA-TSEQLVPAGRGSYIFSQLQAGTNYTLALSMINKQGEGP 520
                 78999999999*****************************9998.**********************************9997 PP
```

The alignment shows the secondary structure of the profile, the model (fn3) and the sequence aligned (7LESS_DROME) with the matches between both, as in PSI-BLAST. The bottom line shows the % of expected accuracy of the match (7 is 65-75%, 8 is 75-85%, 9 is 85-95%, * is 95-100%)

The difference with respect to psi-blast is that we need to know the matrix (profile) of search. This means that when we only have a target sequence we cannot use the approach. Next question is then, what do we do when we only know the target sequence? One way to answer this question is by transforming it into a different one: Having a target sequence, can we have a method to assign the best family profile?

First, we need a database of profiles. How can this database be generated? The initial MSAs of the families are required. These are named seed-MSAs. As an example, we use the alignments of fn3.sto (transcription factor FN3) and pkinase.sto (one family of kinases) as seeds

```
>hmmbuild fn3.hmm fn3.sto
>hmmbuild pkinase.hmm pkinase.sto
>cat globins4.hmm fn3.hmm pkinase.hmm> minifam
```

In order to check sequences and profiles very fast, we compress and index the file with the command:

- **hmmpress [database]**

We run then

```
>hmmpress minifam
```

Now we can search what's the best profile for a given target sequence using the command hmmscan:

- **hmmscan (options) [Database_HMM] [sequence] > [output]**

For example we can use the sequence of 7LES_DROME to search in the database previously generated:

```
>hmmscan minifam 7LES_DROME > 7LES_DROME_minifam.out
```

When looking at the output we can see that the target sequence may fit with more than one profile and more than once with the same profile. This is because sequences can have more than one domain, and in fact each profile is considered a domain, formed by similar sequences.

Finally, the alignment of several sequences can also be done using the HMM profile. This represents a relevant improvement with respect to the alignment of sequences, such as clustalw, for two reasons: 1) the control of gaps and substitutions is referred to the initial seed alignment, and this can be guided by the user (i.e. using structural and/or functional knowledge about the family); and 2) it is faster than any agglomerative approach (i.e. clustalw), as it aligns all sequences with the profile at the same time. The program is hmmalign:

- **hmmalign [model_HMM] [file_with_sequences] > [output]**

We can show this with the file globins45.fa.

Run the following commands and test the speed of both approaches, hmmalign and clustalw:

```
>hmmalign globins4.hmm globins45.fa > globins45.aln
```

The format of the file contains additional rows showing the accuracy of the alignment. This can be transformed into clustalw format running the script:

```
>aconvertMod2.pl –in h –out c <globins45.aln>globins45.clu
```

To test the resulting alignment with clustalw you have to run clustalw with the file globins45.fa (see step 3 of the tutorial in exercise 2.1). However, the question in practice 2.1 was: “Given a sequence, what are the homologs of this sequence in a particular database? (i.e. in a database of sequences with known structure)” . To answer this question we used BLAST and PSI-BLAST. In the package of HMMER3 we also have similar programs: **phmmer** and **jackhmmer**.

The commands to run these programs are:

- **phmmer [target_sequence] [database_of_sequences] > [output]**
- **jackhmmer [target_sequence] [database_of_sequences] > [output]**

The output has the same format as for hmmsearch, and the input are sequences with FASTA format. The difference between phmmer and jackhmmer is that jackhmmer perform several iterations (up to a maximum of five by default, which can be changed using the –N flag) and it generates internally a HMMER profile at each iteration, while phmmer is like jackhmmer at iteraton 1. Iterations are shown in the output entitld by “Round” (search this word in the file to compare the results at different iterations). The internal HMMER profile is used for the search (like with hmmsearch) and the profile is generated with the first sequences matched with the target and taken from the database. We can show this with the file globins45.fa and a globin sequence (i.e. hbb_human).

The problem of this approach is that if the database of sequences where the search is performed is small, the profiles are biased (i.e. searching in “pdb_seq”). We already tackled this problem in “TUTORIAL Step 2” of practice 2.1. How the to solve the problem?

The option would be to find the best HMMER profile in PFAM for our target sequence. PFAM is a database that contains the profiles of several known families of sequences (see BOX). Select the best profile of your choice and fetch it from the database. Now we can run a search of sequences in pdb_seq using this profile. These options are done with the commands **hmmscan** (to scan in PFAM) and **hmmfetch** (to fetch a profile from PFAM).

Step 1) Assign the best profile(s) to the target sequence (i.e. hbb_human):

```
>hmmscan /mnt/NFS_UPF/soft/databases/pfam/Pfam-A.hmm target.fa> target_pfam.out
```

Next, we need to know the HMM of the profile(s) assigned to our target sequence. They can be extracted from the PFAM database using the program hmmfetch:

- **hmmfetch [base_HMM] [name_HMM] > [file_HMM]**

Therefore, in our example, assuming we have found a domain_target:

Step 2) extract the profile(s) from PFAM that correspond to the domains of the target sequence which are found in the column indicated as “model” (see example for fn3 and pkinase as before in minifam). Let’s assume the name of the model we have found for hbb_human is “domain_hbb”, then we execute the command:

```
>hmmfetch /mnt/NFS_UPF/soft/databases/pfam/ Pfam-A.hmm [domain_hbb] > [domain_hbb].hmm
```

Finally we can answer the question:

Step 3) Search for sequences with known structure that contain the same domain as our target

```
>hmmsearch domain_hbb.hmm /mnt/NFS_UPF/soft/databases/blastdat/pdb_seq > hbb_pdb_by_HMM.out
```

Additionally, we have seen how to use aconvertMod2.pl to transform a STOCHOLM format in CLUSTALW. We also need other programs and scripts to transform some of these formats:

1) Transforming a CLUSTALW format of a MSA (Multiple Sequence Alignment) in a STOCKHOLM format. To do this transformation first we need to transform into a FASTA format the alignment using aconvertMod2.pl

```
>aconvertMod2.pl –in c –out f <alignment.clu>alignment.fa
```

Then we transform the FASTA alignment to STOCKHOLM

```
>fasta2sto.pl alignment.fa  > alignment.sto
```

2) We can also transform a STOCKHOLM format MSA file in a FASTA MSA file

```
>sto2fasta.pl –g alignment.sto > alignment.fa
```

3) Transform HMMER3.0 format files in HMMER2.0 format files (old) and ASCII to binary and viceversa.

```
hmmconvert [options] [name_HMM] > [name2_HMM]
```

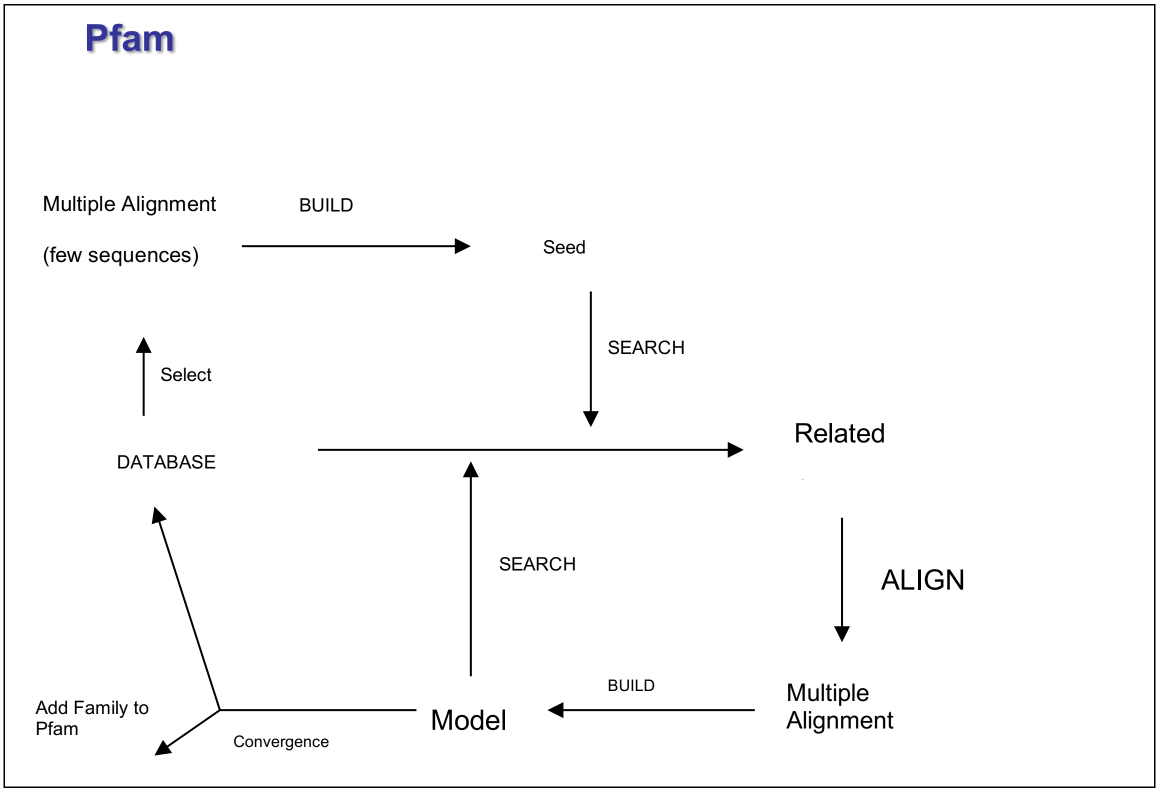
4) Or generate a set of sequences derived from a profile

```
hmmemit [options] –N #number [name_HMM] –o [output]
```

Where the file output contains the generated sequences in FASTA format and #number is the number of sequences to be generated

BOX: PFAM generation

After many years work of biochemists and bioinformaticians several databases of HMM profiles have appeared. The most well known is PFAM, a database of domains of protein families. PFAM has two main groups of families: Pfam-A, formed with the seeds of those very well known families, some of them having a known structure; and Pfam-B, formed with the rest of families which function is not well described and without known structure. The procedure to obtain PFAM A and B is automatic:



QUESTIONS FROM THE TUTORIAL

1) Compare the results of phmmer, jackhmmer with the results of hmmsearch using "domain_hbb.hmm" (see hbb_pdb_by_HMM.out) when searching homologs in pdb_seq for hbb_human.

2) If a protein sequence has more than one domain in PFAM, do you think the result of using hmmsearch and jackhmmer will be the same? Why? Test the example with 7LES_DROME in SwissProt.

3) In practice 2.1 we used PSI-BLAST to fish sequences in the database uniprot_sprot.fasta and generate a PSSM profile which was used for searching homologs in PDB. Check the manual of HMMER3.0 and create your own protocol in which you use the program jackhmmer in a similar approach: use SwissProt database to generate the HMM profile and perform the search in pdb_seq.

4) Use hmmscan to search the best model(s) for 7LES_DROME in PFAM and search the homologs in PDB with this/these model(s). Compare the results of this search with the results of your protocol search in question 3. What are the differences? Why?

5) Use your protocol described in question 3 to search homologs of 7LES_DROME in PDB and compare with the results of the protocol described in practice 2.1 when using PSI-BLAST.

6) Use the sequence target.fa from practice 2.1. Apply phmmer, jackhmmer and the protocols of questions 3 and 4 to find homologs in PDB. What's the fold of this sequence? Compare the result with the homologs found in practice 2.1

7) Use hmmalign and FetchFasta.pl to align the sequence of target.fa and its homologs of PDB

8) If you have to align the sequence 7LES_DROME and its homologs of PDB what's the best model to use? Produce the alignment with the models from question 4 and your protocol in question 3 to show your answer.

9) What are the folds of the following sequences?

   a. *problem1/serc_myctu.fa*

   b. *problem2/p72_mycmy.fa*

   c. *problem3/lip_staau.fa*

   d. *problem4/orc1_human.fa*